



# Módulos Temporizadores



## CARACTERÍSTICAS GENERALES

### Temporizador TMR0

Contador / Temporizador de 8 bits que se pueden leer y escribir.  
Prescaler de 8 bits programable por software.  
Generación de interrupción al desbordarse de FFh a 00h.  
Se puede seleccionar el flanco activo si se usa reloj externo.

### Temporizador TMR1

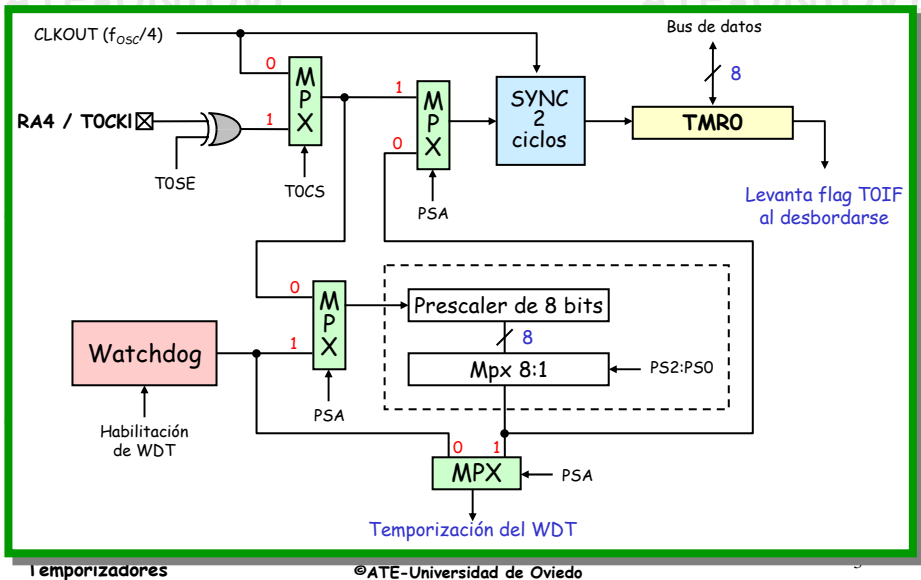
Contador / Temporizador de 16 bits que se pueden leer y escribir.  
Prescaler programable por software.  
Generación de interrupción al desbordarse de FFFFh a 0000h.  
Puede activarse o desactivarse.

### Temporizador TMR2

Temporizador de 8 bits que se pueden leer y escribir.  
Prescaler para el reloj y postscaler para la salida.  
TMR2 se incrementa hasta alcanzar el valor de PR2.  
Puede activarse o desactivarse.



## TEMPORIZADOR TMRO/WDT



## Registro OPTION REG (81h, 181h)

RBPV	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
------	--------	------	------	-----	-----	-----	-----

bit 7 **RBPV**

bit 6 **INTEDG**

bit 5 **TOCS**: Selección del reloj a utilizar

0: Reloj interno ( $f_{clk}/4$ )

1: Reloj externo (RA4)

bit 4 **TOSE**: Selección del flanco activo del reloj externo

0: Flanco de subida

1: Flanco de bajada

bit 3 **PSA**: Asignación del prescaler

0: Asignado a TMRO

1: Asignado a WDT

bits 2:0 **PS2:PS0**: Prescaler

000: 1:2 para TMRO / 1:1 para WDT.

001: 1:4 para TMRO / 1:2 para WDT.

010: 1:8 para TMRO / 1:4 para WDT.

011: 1:16 para TMRO / 1:8 para WDT.

100: 1:32 para TMRO / 1:16 para WDT.

101: 1:64 para TMRO / 1:32 para WDT.

110: 1:128 para TMRO / 1:64 para WDT.

111: 1:256 para TMRO / 1:128 para WDT.



### Registro TMRO (01h, 101h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------

\* Muestra en todo momento el número de pulsos que se han producido.

Cuenta desde 0 (0x00) hasta 255 (0xFF).

La cuenta es cíclica: tras el 255 se vuelve al 0 (desbordamiento).

Si cuenta pulsos del oscilador, está constantemente incrementándose.

### Registro INTCON (0Bh, 8Bh, 10Bh, 18Bh)

GEIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
------	------	------	------	------	------	------	------

\* Se puede generar una interrupción cada vez que se desborda TMRO.

Es preciso habilitar la interrupción por *overflow* de TMRO (TOIE=1).

También hay que habilitar la máscara global de interrupciones (GEIE=1).

Aun sin habilitar la interrupción, TOIF siempre se pone a 1 tras un *overflow* de TMRO.

El usuario debe encargarse de hacer TOIF=0 en el programa.



### Cálculo de la temporización en TMRO

$$\text{temp} = [(256 - \text{carga}) \cdot \text{PS} + 2] \cdot T_{\text{instr}}$$

Temporización

Incrementos hasta desbordamiento

Si se asigna a WDT tomaría el valor 1

Ciclos de sincronización tras cargar TMRO

$4 / f_{\text{osc}}$

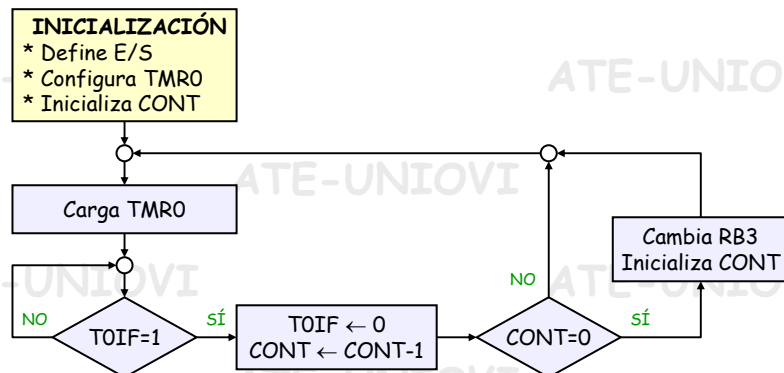
• Se llama **carga** al valor que se asigna al registro TMRO (01h, 101h) al comenzar la temporización.

• El valor del *prescaler* **PS** queda determinado por los tres bits más bajos del registro OPTION\_REG (81h, 181h). Si el *prescaler* se le asigna al *watchdog* (bit PSA=1), se tiene **PS**=1 para esta expresión.



### Ejemplo 1 (*temp01.asm*)

- Fichero incluido con el curso. Utiliza el temporizador TMRO para hacer que el LED conectado a RB3 parpadee: 500ms encendido y 500ms apagado.



### EJERCICIO

**Comprobar el funcionamiento del programa *temp01.asm***

- Crear un proyecto con dicho fichero.
- Emular el comportamiento del programa y verificar que el LED se enciende según lo previsto.
- Visualizar el contenido de algunos registros usando una ventana de visualización (*Watch*).
- Introducir las modificaciones necesarias para que el LED esté 500ms encendido y 1s apagado.

ATE-UNIOVI



9

ATE-UNIOVI      Registro T1CON (10h)      ATE-UNIOVI

ATE-UNTOVT

---

10



### Registros TMR1L (0Eh) y TMR1H (0Fh)

7								0									7												0
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0													
TMR1H									TMR1L																				

\* Muestran en todo momento el número de pulsos que se han producido.

Cuenta desde 0 (0x0000) hasta 65.535 (0xFFFF).

La cuenta es cíclica: tras el 65.535 se vuelve al 0 (desbordamiento).

Puede apagarse.

### Registro INTCON (0Bh, 8Bh, 10Bh, 18Bh)

GEIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
------	------	------	------	------	------	------	------

\* Se puede generar una interrupción cada vez que se desborda TMR1.

La habilitación de la interrupción correspondiente se hace en otro registro.

En este registro hay que habilitar la máscara global de interrupciones (GEIE=1).

También se habilita el registro de interrupciones generadas por periféricos (PEIE=1).



### Registro PIE1 (8Ch)

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
-------	------	------	------	-------	--------	--------	--------

\* Contiene los bits de habilitación de varias interrupciones generadas por periféricos.

Se puede generar una interrupción cada vez que se desborda TMR1.

Para ello hay que habilitar dicha interrupción (TMR1IE=1).

### Registro PIR1 (0Ch)

PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
-------	------	------	------	-------	--------	--------	--------

\* Contiene los *flags* de las interrupciones contenidas en PIE1.

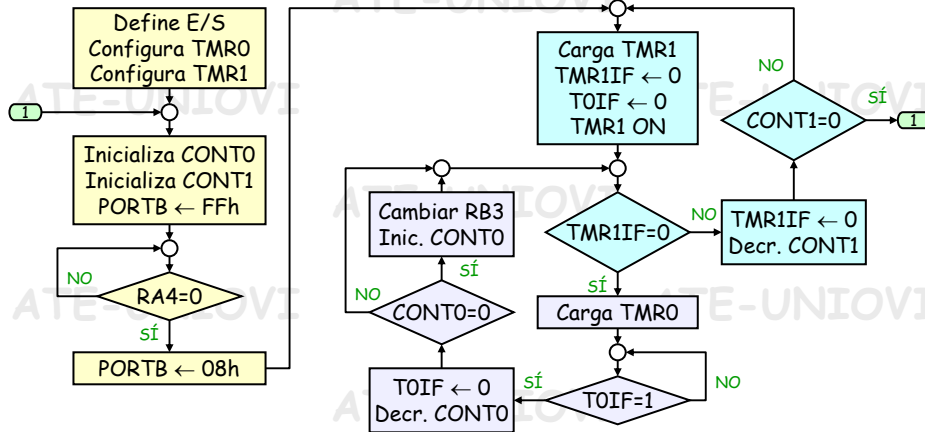
Aun sin habilitar la interrupción, TMR1IF siempre se pone a 1 tras un *overflow* de TMR1.

El usuario debe encargarse de hacer TMR1IF=0 en el programa.



**Ejemplo 2 (temp02.asm)**

- Fichero incluido con el curso. Utiliza el temporizador TMRO para hacer que el LED conectado a RB3 parpadee: 500ms encendido y 500ms apagado. Este parpadeo sólo se produce durante 5s (temporizados con TMR1).



Temporizadores

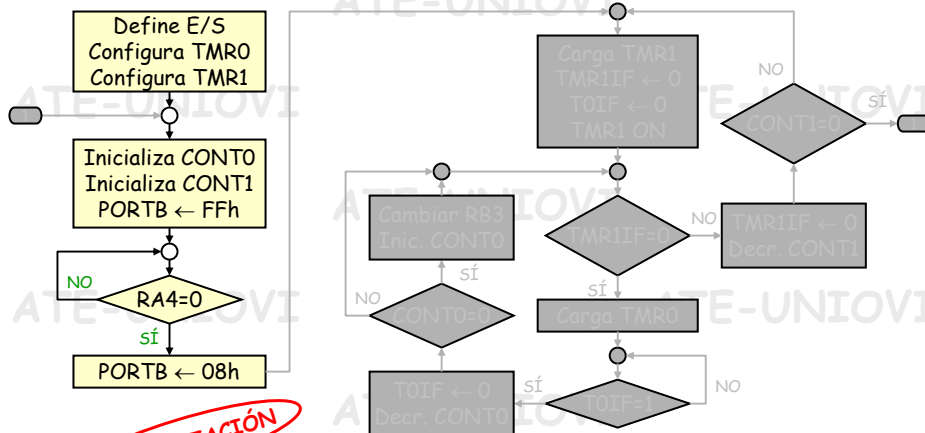
©ATE-Universidad de Oviedo

13



**Ejemplo 2 (temp02.asm)**

- Fichero incluido con el curso. Utiliza el temporizador TMRO para hacer que el LED conectado a RB3 parpadee: 500ms encendido y 500ms apagado. Este parpadeo sólo se produce durante 5s (temporizados con TMR1).



Tem

**INICIALIZACIÓN**

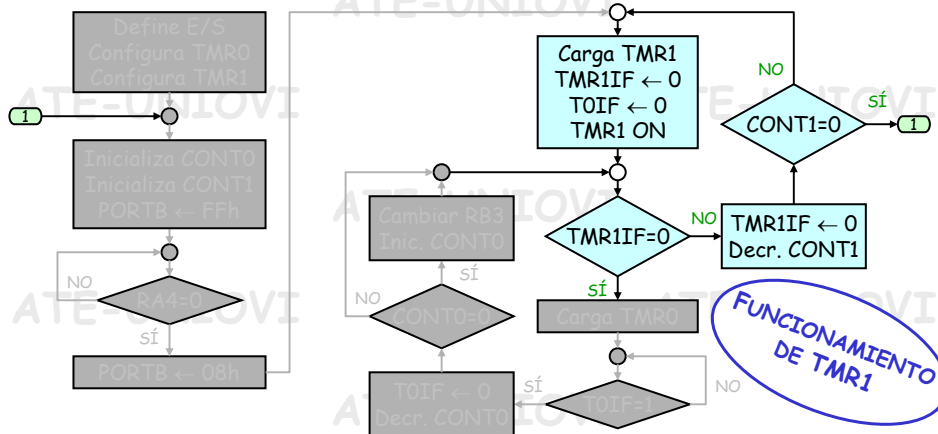
©ATE-Universidad de Oviedo

14



### Ejemplo 2 (temp02.asm)

- Fichero incluido con el curso. Utiliza el temporizador TMRO para hacer que el LED conectado a RB3 parpadee: 500ms encendido y 500ms apagado. Este parpadeo sólo se produce durante 5s (temporizados con TMR1).



Temporizadores

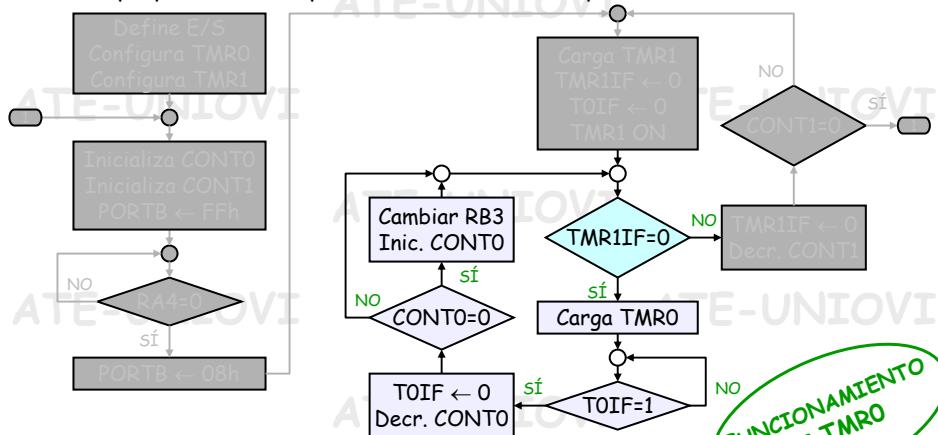
©ATE-Universidad de Oviedo

15



### Ejemplo 2 (temp02.asm)

- Fichero incluido con el curso. Utiliza el temporizador TMRO para hacer que el LED conectado a RB3 parpadee: 500ms encendido y 500ms apagado. Este parpadeo sólo se produce durante 5s (temporizados con TMR1).



Temporizadores

©ATE-Universidad de Oviedo





### EJERCICIO

*Comprobar el funcionamiento del programa  
temp02.asm*

- Crear un proyecto con dicho fichero.
- Emular el comportamiento del programa y verificar que el LED se enciende según lo previsto.
- Visualizar el contenido de algunos registros usando una ventana de visualización (*Watch*).
- Introducir las modificaciones necesarias para que el LED esté parpadeando 7s a una velocidad mayor que la dada inicialmente.



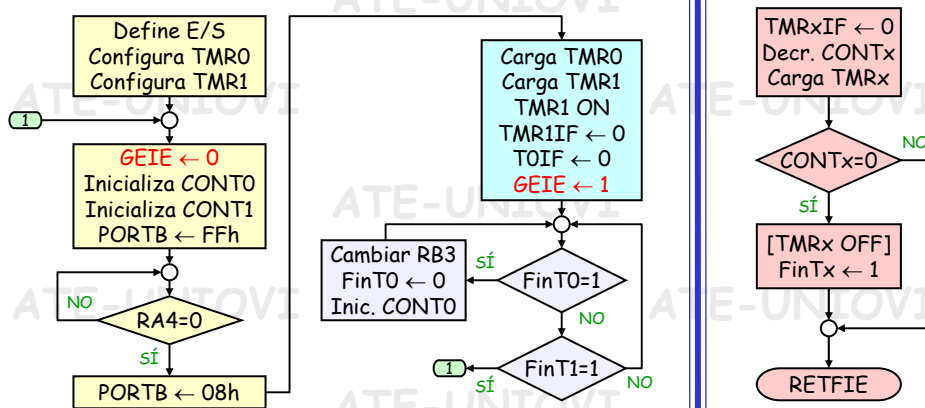
### UTILIZACIÓN de INTERRUPCIONES

- Es muy habitual que en los programas las **temporizaciones** se lleven a cabo **mientras se están ejecutando otras tareas**.
- En estos casos, es preferible dejar **que sea el propio microcontrolador el que nos avise** de cuándo ha finalizado la temporización, sin que el programa principal tenga que estar verificando periódicamente si se ha levantado el **flag** correspondiente.
- El servicio de una interrupción producida por desbordamiento de uno de los temporizadores del microcontrolador deberá encargarse de:
  - a) **Guardar el estado** del sistema (W, STATUS, PCLATH).
  - b) **Borrar el flag** de la interrupción.
  - c) Generar los **cambios** que permitan que el programa principal pase a otro modo de funcionamiento.
  - d) Si la temporización precisa de varios desbordamientos del temporizador, **recargar el temporizador** y **actualizar la cuenta** de **overflows** producidos.



### Ejemplo 3 (*temp03.asm*)

- Fichero incluido con el curso. Hace uso de las interrupciones asociadas a TMR0 y TMR1 para repetir el funcionamiento de *temp02.asm*.



### EJERCICIO

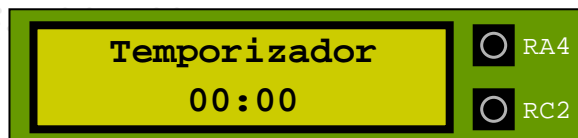
**Comprobar el funcionamiento del programa *temp03.asm***

- Crear un proyecto con dicho fichero.
- Emular el comportamiento del programa y verificar que el LED se enciende según lo previsto.
- Visualizar el contenido de algunos registros usando una ventana de visualización (*Watch*).
- Introducir las modificaciones necesarias para que el LED esté parpadeando 7s a una velocidad mayor que la dada inicialmente.



### EJERCICIO PROPUESTO

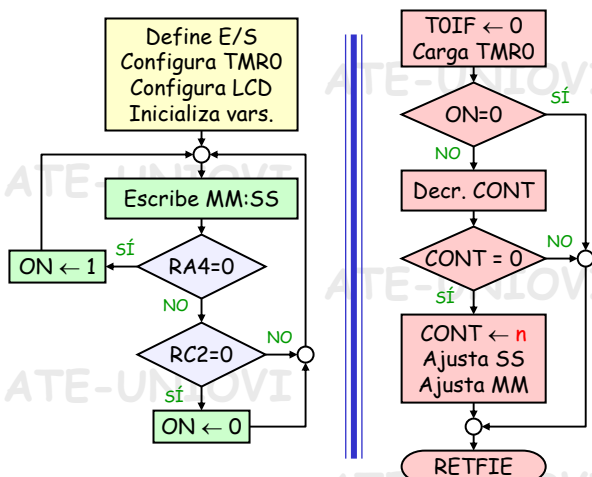
- Desarrollar un cronómetro digital capaz de temporizar 60 minutos en cuenta ascendente.
- El cronómetro arrancará parado mostrando 00:00 y sólo se pondrá en marcha si se acciona el pulsador RA4. A partir de ese momento, la cuenta se irá incrementando de segundo en segundo.
- Si durante el funcionamiento del cronómetro se pulsa el botón RC2(\*), la cuenta se detendrá. Tras pulsar de nuevo RA4, la temporización continúa desde el punto en que se había detenido.
- La cuenta debe ser cíclica: al minuto 59 debe seguirle el minuto 00.



(\*) RBO en placas PICDEM 2 PLUS



### Algunas ideas

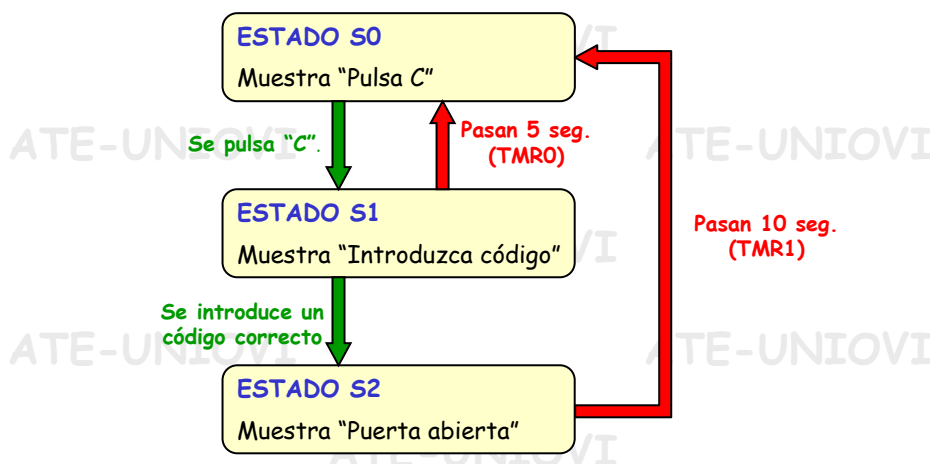


- Activar **interrupciones** de TMRO.
- Cada 1/n segundos se genera una interrupción.
- Tras n interrupciones, hay que aumentar en una unidad la cuenta de segundos.
- Puede que sea necesario **ajustar** las decenas de segundo y/o los minutos.
- Fuera de la interrupción, el programa se limita a **escribir la hora** y a **explorar las teclas**.



### EJERCICIO ALTERNATIVO

- Completar el funcionamiento del sistema de identificación de claves.



### Algunas ideas

- En el estado S1, además de leer las teclas para determinar si el código introducido es el correcto, se cargará TMR0 con el valor adecuado y se habilitará su interrupción.
- En el estado S2 se deshabilita la interrupción por desbordamiento del temporizador TMR0.
- También en este estado S2 se encenderá TMR1 tras haberlo cargado con el valor adecuado para llevar a cabo la temporización deseada. La interrupción asociada a TMR1 debe estar habilitada.
- Asegurarse de que en el estado S0 no está habilitada ninguna de las interrupciones asociadas a los temporizadores.

